

09/601332



BREVET D'INVENTION

REC'D 15 FEB 1999

WIPO PCT

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 28 JAN. 1999

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS Cédex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30

THIS PAGE BLANK (USPTO)

REQUÊTE EN DÉLIVRANCE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

Confirmation d'un dépôt par télécopie ☐

Cet imprimé est à remplir à l'encre noire en lettres capitales

Réservé à l'INPI

DATE DE REMISE DES PIÈCES **29 JAN 1998**
N° D'ENREGISTREMENT NATIONAL **98 01008 -**
DÉPARTEMENT DE DÉPÔT **X**
DATE DE DÉPÔT **29 JAN, 1998**

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE
À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

CABINET BALLOT-SCHMIT
7, rue Le Sueur
75116 PARIS
FRANCE

2 DEMANDE Nature du titre de propriété industrielle

☒ brevet d'invention

☐ demande divisionnaire

☐ certificat d'utilité

☐ transformation d'une demande
de brevet européen

☐ demande initiale

☐ brevet d'invention

n° du pouvoir permanent

références du correspondant

téléphone

SM/ 013815 01.40.67.11.99

☐ certificat d'utilité n°

date

Établissement du rapport de recherche

☐ différé

☒ immédiat

Le demandeur, personne physique, requiert le paiement échelonné de la redevance

☐ oui

☒ non

Titre de l'invention (200 caractères maximum)

SYSTEME ET PROCEDE DE GESTION DE SECURITE D'APPLICATIONS INFORMATIQUES.

3 DEMANDEUR (S) n° SIREN **7 4 9 7 1 1 2 0**

code APE-NAF

Nom et prénoms (souligner le nom patronymique) ou dénomination

GEMPLUS

Forme juridique

Société en
Commandite par
par Actions

Nationalité (s) **Française**

Adresse (s) complète (s)

Pays

Avenue du Pic de Bertagne
Parc d'activités de la Plaine de Jouques
13420 GEMENOS

FRANCE

En cas d'insuffisance de place, poursuivre sur papier libre

4 INVENTEUR (S) Les inventeurs sont les demandeurs

☐ oui

☒ non

Si la réponse est non, fournir une désignation séparée

5 RÉDUCTION DU TAUX DES REDEVANCES

☐ requise pour la 1ère fois

☐ requise antérieurement au dépôt : joindre copie de la décision d'admission

6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE

pays d'origine

numéro

date de dépôt

nature de la demande

7 DIVISIONS antérieures à la présente demande n°

date

n°

date

8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE

(nom et qualité du signataire - n° d'inscription)

Paul BALLOT
92-1009
CABINET BALLOT SCHMIT

SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION

SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI

[Signature]

DÉSIGNATION DE L'INVENTEUR

(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

DIVISION ADMINISTRATIVE DES BREVETS

26bis, rue de Saint-Petersbourg
75800 Paris Cédex 08

Tél. : 01 58 94 53 94 Télécopie : 01 42 93 59 30

BN/015815

N° D'ENREGISTREMENT NATIONAL

75116

TITRE DE L'INVENTION :

**SYSTEME ET PROCEDE DE GESTION DE SECURITE D'APPLICATIONS
INFORMATIQUES.**

LE(S) SOUSSIGNÉ(S)

Cabinet BALLOT SCHMIT
7, rue Le Sueur
75116 PARIS
FRANCE

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

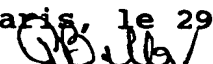
COULIER Charles
BRUN Philippe

domicilié (s) au :

Cabinet BALLOT SCHMIT
7, rue Le Sueur
75116 PARIS
FRANCE

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire

Paris, le 29 janvier 1998

Paul BALLOT 92-1009
Cabinet BALLOT SCHMIT

DOCUMENT COMPORTANT DES MODIFICATIONS

PAGE(S) DE LA DESCRIPTION OU DES REVENDICATIONS OU PLANCHE(S) DE DESSIN			R.M.*	DATE DE LA CORRESPONDANCE	TAMPON DATEUR DU CORRECTEUR
Modifiée(s)	Supprimée(s)	Ajoutée(s)			
152'19			RM	30 04 98	06 1998 - A N R

Un changement apporté à la rédaction des revendications d'origine, sauf si celui-ci découle des dispositions de l'article 28 du décret du 19 septembre 1979, est signalé par la mention "R.M." (revendications modifiées).

SYSTEME ET PROCEDE DE GESTION DE SECURITE D'APPLICATIONS INFORMATIQUES

L'invention concerne les systèmes informatiques et, plus particulièrement dans de tels systèmes, un système et procédé pour gérer les conditions d'accès aux différentes applications qui sont susceptibles d'être
5 mises en oeuvre par ces systèmes informatiques. L'invention est préférentiellement, mais non limitativement, destinée à être mise en oeuvre dans les microprocesseurs des cartes à puce quel que soit le domaine d'utilisation: santé, banque, transport,
10 téléphone mobile etc ...

Les procédés connus de gestion de sécurité présentent les principaux inconvénients suivants:

- le premier inconvénient est une obligation de présenter une hiérarchie pour sélectionner une
15 application, c'est-à-dire qu'il faut passer par un chemin de sélection imposé en commençant par l'application "grand-mère", puis l'application "mère" pour arriver à l'application "fille", c'est-à-dire un chemin de sélection analogue à celui pour sélectionner
20 un fichier dans un répertoire d'un disque dur ; en outre, il n'y a rien de prévu au sujet de la sécurité.

Il n'existe donc pas de relation entre le niveau de la sélection et celui de la sécurité.

- Le second inconvénient est de limiter le nombre
25 de niveaux de sécurité ou le nombre d'applications. En effet, à chaque application est dédié un registre de sécurité qui mémorise les droits acquis par cette application par la connaissance de secrets. Pour rajouter n niveaux, c'est-à-dire disposer d'une série
30 multi-applicative, il faut associer, par exemple, un registre de sécurité à chaque application, ce qui

conduit à utiliser une partie importante de la mémoire rapide où sont stockés les registres de sécurité. Comme la capacité de cette mémoire rapide est limitée, il n'est pas souhaitable d'y stocker de nombreux registres de sécurité. C'est ainsi que dans certains systèmes, le
5 nombre de niveaux hiérarchiques ou le nombre d'applications a été limité à trois, soit trois registres de sécurité.

- Le troisième inconvénient est d'empêcher
10 "l'émancipation" simple des applications, c'est-à-dire rendre une application "fille" indépendante de son application "mère". En effet, lors d'une création d'une nouvelle application, il est indispensable d'utiliser les droits et secrets de l'application "mère" qui sont
15 les seuls disponibles et ce jusqu'à la création des secrets propres à l'application "fille".

Le but de la présente invention est de mettre en oeuvre un procédé de gestion de sécurité d'applications informatiques qui ne présente pas les inconvénients
20 exposés ci-dessus et qui permet donc:

- de ne pas être limité en nombre de niveaux hiérarchiques ou nombre d'applications, et
- de rendre une application "fille" indépendante de l'application "mère" sans passer par cette dernière du
25 point de vue de la sécurité.

L'invention concerne donc un système de gestion de la sécurité d'applications informatiques, caractérisé en ce que:

- les applications informatiques sont enregistrées
30 dans des fichiers répertoires organisés suivant une arborescence à n niveaux, le répertoire de niveau 1 étant de niveau le plus élevé, et

- un nombre r de registres de sécurité pouvant être affectés chacun à un seul répertoire et chaque registre

de sécurité contenant l'ensemble des droits ou secrets S1 à Sp qui ont été octroyés sous un répertoire.

L'invention concerne également un procédé de gestion de la sécurité d'applications informatiques dans le système de gestion décrit ci-dessus, caractérisé en ce qu'il comprend les étapes suivantes consistant à:

(a) mémoriser dans des registres de sécurité les droits octroyés sous un répertoire selon des règles déterminées,

(b) rechercher dans l'arborescence les secrets présentés, et

(c) vérifier la connaissance équivalente à un (ou des) droits au niveau de l'application informatique pour satisfaire les conditions d'accès.

D'autres caractéristiques et avantages de la présente invention apparaîtront à la lecture de la description suivante d'exemples particuliers de réalisation, ladite description étant faite en relation avec les dossiers joints dans lesquels:

- la figure 1 est un exemple de structure arborescente des répertoires;

- les figures 2.1 à 2.14 illustrent des exemples d'application des trois règles d'attribution ou de désaffectation d'un registre de sécurité à un répertoire;

- les figures 3.1a à 3.6a et 3.1b à 3.6b illustrent des exemples d'application de la règle de présentation d'un secret; et

- les figures 4.1 à 4.6 illustrent des exemples d'application de la règle de vérification de l'octroi du droit requis.

L'invention sera décrite dans son application à une carte à puce et, plus précisément, à un microprocesseur utilisé dans une carte à puce. Cependant, elle est

également applicable à tout système informatique où il est nécessaire ou simplement souhaitable que certains services ou fonctions offerts par le système soient accessibles seulement à certains utilisateurs ou opérateurs.

Dans le cas de cartes à puce, par exemple une carte bancaire ou une carte de téléphone mobile, les services ou fonctions qui sont à la disposition de l'utilisateur peuvent être soumis à autorisation selon le type d'abonnement souscrit, ces autorisations (ou droits) étant octroyées en prouvant la connaissance de secrets qui permettent l'accès aux fichiers nécessaires à la mise en oeuvre du service ou de la fonction.

Dans la suite de la description, les définitions suivantes seront adoptées:

- Un fichier est un ensemble de données pouvant être protégées par des conditions d'accès.
- Un répertoire Rep est un ensemble de fichiers et/ou de répertoires selon une organisation arborescente (figure 1); habituellement un répertoire est dédié uniquement à une application.
- Les conditions d'accès à un fichier ou à un répertoire Rep définissent les critères à remplir, telle que la présentation d'un code secret ou une authentification externe, pour pouvoir effectuer telle ou telle fonction sur le fichier ou le répertoire;
- Les fichiers et répertoires sont organisés suivant une arborescence à plusieurs niveaux dont le répertoire de plus haut niveau (niveau 1) est appelé "répertoire racine" ou racine de l'arborescence. Un niveau caractérise les répertoires ayant le même degré hiérarchique. L'utilisation de répertoires permet de structurer les données d'une carte à puce. Sur la figure 1, seuls les répertoires Rep1, Rep2, Rep31,

Rep32, Rep41, Rep42, Rep51 et Rep52 ont été présentés et chacun peut contenir un ou plusieurs fichiers. Le répertoire Rep1 est la racine de l'arborescence comprenant $n=5$ niveaux de répertoires, les répertoires Rep41 et Rep42 appartenant au niveau $i=4$.

- Un registre de sécurité R contient l'ensemble des droits qui ont été octroyés sous un répertoire et un droit est la preuve de la connaissance d'un secret qui est identifié par une référence telle qu'un nom, un numéro, un identificateur. Il y a plusieurs façons de prouver la connaissance d'un secret, par exemple par échange de la valeur du secret entre le terminal et la carte à puce ou par échange de données calculées à l'aide de ce secret: l'opération s'appelle présentation du secret.

D'une manière générale, le fondement de la sécurité sur une carte à puce est de pouvoir subordonner l'utilisation du service ou de la fonction de la carte à puce à la preuve de la connaissance d'un ou plusieurs secrets. Ainsi pour pouvoir utiliser une fonction de la carte, il faut:

- que la carte à puce mémorise préalablement cette preuve de la connaissance du ou des secrets dans un registre de sécurité,

- que le porteur de la carte à puce ou le terminal prouve qu'il a connaissance du (ou des) secret(s) protégeant la fonction,

- que la carte vérifie, lors de l'utilisation de la fonction, que le (ou les) secret(s) est (sont) bien connu(s) .

L'invention réside dans les étapes du procédé consistant à:

(a) mémoriser dans la carte à puce la connaissance du (ou de(s)) secret(s) c'est-à-dire les droits

octroyés, selon des règles d'attribution et de désaffectation d'un registre de sécurité à un répertoire,

(b) rechercher dans l'arborescence le (ou les) secret(s) présenté(s),

5 (c) vérifier la connaissance du (ou des) secret(s) pour remplir les conditions d'accès.

Pour mémoriser la connaissance d'un secret dans une carte à puce (étape (a)), il est nécessaire de présenter correctement le secret, ce qui revient à prouver que l'extérieur, par exemple un terminal ou un porteur de carte, a la connaissance dudit secret, cette connaissance lui conférant un droit d'utilisation de fonctions ou de services offerts par la carte. C'est le droit qui est mémorisé dans un registre de sécurité à raison d'un registre par application.

Un registre de sécurité R comprend un nombre p de chiffres ou positions, chaque position étant affectée à la connaissance d'un secret correspondant à un droit octroyé. Un registre à p=8 positions pourra enregistrer la connaissance de huit secrets S1 à S8 qui correspondront à huit droits octroyés.

Le nombre r de registres de sécurité R peut être quelconque et l'exemple qui sera décrit en comportera r=3. Les registres de sécurité ne sont pas dédiés à un niveau ou à un répertoire donné comme dans l'art antérieur et le lien entre un répertoire et un registre de sécurité est dynamique, c'est-à-dire que ce lien peut être créé ou rompu conformément aux règles du procédé selon l'invention.

30 Pour mémoriser un droit dans un répertoire, il faut d'abord attribuer ou désaffecter un registre de sécurité à un répertoire selon les trois règles RG1 à RG3 suivantes:

Règle RG1:

Un registre est attribué au répertoire courant dès lors qu'un droit est octroyé sous ce répertoire, par exemple un code secret ou une authentification. Si un droit a déjà été octroyé sous ce répertoire, le
5 registre dédié à celui-ci est mis à jour.

Règle RG2:

La sélection d'un nouveau répertoire entraîne la perte du lien reliant l'ancien répertoire courant à son registre de sécurité sauf si le répertoire sélectionné
10 est "fils" de l'ancien répertoire courant.

Règle RG3:

Si le nombre r de registres de sécurité est saturé, c'est-à-dire que les $r=3$ de l'exemple décrit sont utilisés, le registre le plus anciennement affecté,
15 c'est-à-dire le niveau le plus haut dans l'arborescence, est attribué au nouveau répertoire courant conformément à la règle RG1.

Il est à remarquer que l'application de la règle RG2 rend impossible l'attribution de deux registres de
20 sécurité à un même niveau, de sorte que l'attribution d'un registre de sécurité à un répertoire peut être matérialisée par un niveau hiérarchique N_i affecté au registre de sécurité concerné, i variant de 1 à n .

Les figures 2.1 à 2.14 illustrent des applications
25 des règles RG1, RG2 et RG3. Sur ces figures et les autres, un cercle noir désigne un répertoire, un cercle gris désigne un répertoire sélectionné et un cercle blanc désigne un répertoire sélectionné avec un droit levé.

30 La figure 2.1 illustre l'absence de sélection d'un répertoire tandis que les figures 2.2 et 2.3 illustrent respectivement la sélection des répertoires Rep1 et Rep2.

Ainsi, l'application de la règle RG1 est illustrée dans les figures 2.4, 2.6, 2.8, 2.10, 2.12 et 2.14. La figure 2.4 illustre la présentation d'un secret sous le répertoire Rep2 de niveau N2. La figure 2.6 illustre la
5 présentation d'un secret sous le répertoire Rep31 de niveau N3. La figure 2.8 illustre la présentation d'un droit sous le répertoire Rep41 de niveau N4. La figure 2.10 illustre la présentation d'un droit sous le répertoire Rep51 de niveau N5. La figure 2.12 illustre
10 la présentation d'un droit sous le répertoire Rep41. La figure 2.14 illustre la présentation d'un droit sous le répertoire Rep42.

L'application de la règle RG2 est illustrée par les figures 2.5, 2.7 et 2.9 en ce qui concerne le maintien
15 du lien entre un registre de sécurité et son répertoire lors de la sélection d'un nouveau répertoire "fils" de celui-ci.

Les figures 2.5, 2.7 et 2.9 illustrent respectivement la sélection du répertoire Rep31, Rep41
20 ou Rep51.

L'application de la règle RG2 est illustrée par les figures 2.11 et 2.13 en ce qui concerne la rupture du lien entre un registre de sécurité et son répertoire. Ainsi, la figure 2.11 illustre la sélection du
25 répertoire Rep41 tandis que la figure 2.13 illustre la sélection du répertoire Rep42.

L'application de la règle RG3 est illustrée par la figure 2.10 dans laquelle le registre le plus
anciennement affecté R1 est attribué au nouveau
30 répertoire sélectionné Rep51.

L'étape (a) consistant à mémoriser les droits attachés à la connaissance des secrets étant réalisée, l'étape (b) consistant à rechercher dans l'arborescence

le secret présenté par le porteur de la carte à puce ou par le terminal peut être mise en oeuvre.

Un secret présenté au niveau d'une application confère un droit d'utilisation au niveau de cette même application. Ainsi, la présentation réussie d'un secret au sein d'une application de niveau hiérarchique Ni met à jour le registre de sécurité dédié à ce niveau hiérarchique, conformément à la règle RG1, même si le secret présenté est physiquement situé dans un niveau hiérarchique supérieur.

La règle de présentation d'un secret est la suivante:

Règle RG4:

La présentation d'un secret de référence S revient à vérifier que le porteur de la carte à puce ou le terminal connaît la valeur du premier secret de référence S trouvé en parcourant l'axe hiérarchique de l'application courante vers le répertoire racine.

La présentation du secret de référence S au niveau de l'application courante située au niveau hiérarchique Ni est réalisée par les étapes intermédiaires suivantes consistant à:

(b1) rechercher un secret de référence S dans le répertoire courant, c'est-à-dire au niveau Ni, à l'aide du système de gestion de sécurité et vérifier l'existence de ce secret au sein de l'application;

(b2) si ce secret existe, vérifier que la présentation du secret est réussie, par exemple valeur pour un code secret, cryptogramme pour une clé, etc ...,

Si la présentation est réussie, le droit associé au secret de référence S est octroyé au niveau de l'application courante de niveau Ni.

Si la présentation a échoué, le droit associé au secret de référence S n'est pas octroyé et la tentative de présentation est terminée.

5 (b3) Si le secret de référence S n'existe pas au sein de l'application courante de niveau Ni, rechercher si un secret de même référence existe au sein de l'application parente de niveau N(i-1) de l'application courante.

10 (b4) Si le secret existe au niveau de l'application parente de niveau N(i-1), vérifier que la présentation est réussie.

Si la présentation est réussie, le droit associé au secret de référence S est octroyé au niveau de l'application courante de niveau Ni.

15 Si la présentation a échoué, le droit associé au secret de référence S n'est pas octroyé et la tentative de présentation est terminée.

20 (b5) Si le secret de référence S n'existe pas au sein de l'application parente de niveau N(i-1), rechercher le secret de référence S au niveau N(i-2) suivant l'axe hiérarchique, et ainsi de suite tant que l'existence d'un secret de référence S n'a pas été découverte.

25 (b6) Si le secret de référence S n'a pas été trouvé, la tentative de présentation est terminée.

30 Plusieurs exemples d'application de la règle RG4 sont illustrés sur les figures 3.1a à 3.6a et 3.1b à 3.6b. Les figures 3.1a et 3.1b, 3.2a et 3.2b, 3.3a et 3.3b correspondent à des exemples où le droit est octroyé tandis que les figures 3.4a et 3.4b, 3.5a et 3.5b, 3.6a et 3.6b correspondent à des exemples où le droit n'est pas octroyé.

Sur la figure 3.1a, le secret S3 existe en local sous le répertoire Rep41 et aucun registre n'est

attribué au répertoire Rep41. Sur la figure 3.1b, la connaissance du secret S3 est prouvée ; un registre R3 est affecté au répertoire Rep41 de niveau N4 et le droit est octroyé.

5 Sur la figure 3.2a, le secret S3 existe en local sous le répertoire Rep41 et un registre R3 est déjà attribué au répertoire Rep41. La connaissance du secret S3 est donc prouvée et le registre de sécurité R3 affecté au répertoire Rep41 est mis à jour (S3) de
10 sorte que le droit est octroyé (figure 3.2b).

Sur la figure 3.3a, le secret S2 n'existe pas en local sous le répertoire Rep41 ; un registre R3 est déjà attribué au répertoire Rep41 et un secret S2 existe à la fois sous les répertoires Rep2, Rep1, Rep42
15 et Rep51. La connaissance du secret S2 est donc prouvée et le registre de sécurité affecté au répertoire Rep41 est mis à jour de sorte que le droit est octroyé (figure 3.3b).

Sur la figure 3.4a, le secret S2 n'existe pas en
20 local sous le répertoire Rep41 ; un registre R3 est déjà attribué au répertoire Rep41 et un secret S2 existe à la fois sous les répertoires Rep2, Rep1, Rep42 et Rep51. La connaissance du secret S2 n'est donc pas prouvée de sorte que le registre de sécurité R3 affecté
25 au répertoire Rep41 n'est pas mis à jour et que le droit n'est pas octroyé (figure 3.4b).

Sur la figure 3.5a, le secret S2 n'existe pas en local sous le répertoire Rep41 ; un registre R3 est déjà attribué au répertoire Rep41 et un secret S2
30 existe à la fois sous les répertoires Rep2, Rep1, Rep42 et Rep51. La connaissance du secret S2 n'est donc pas prouvée de sorte que le registre de sécurité R3 affecté au répertoire Rep41 n'est pas mis à jour et le droit n'est pas octroyé (figure 3.5b).

Sur la figure 3.6a, le secret S2 n'existe pas en local sous le répertoire Rep41 ; un registre R3 est déjà attribué au répertoire Rep41 et un secret S2 existe à la fois sous les répertoires Rep2, Rep1, Rep42 et Rep51. La connaissance du secret S2 n'est pas prouvée de sorte que le registre de sécurité R3 affecté au répertoire Rep41 n'est pas mis à jour et le droit n'est pas octroyé (figure 3.6b).

L'étape (c) consiste à vérifier que la connaissance du (ou des) secret(s) pour remplir les conditions d'accès, c'est-à-dire vérifier que le secret protégeant l'utilisation d'une fonction et d'un service de la carte à puce est bien connu du monde extérieur, c'est-à-dire que le droit requis a bien été octroyé.

A cet effet, l'invention met en oeuvre une cinquième règle RG5 qui s'énonce de la manière suivante:

Règle RG5:

Une fonction, nécessitant la connaissance d'un secret S, est autorisée si et seulement si, en parcourant l'arborescence suivant l'axe hiérarchique de l'application courante vers l'application racine, le premier secret S rencontré est connu, c'est-à-dire correctement présenté, par au moins l'une des applications appartenant à la section arborescente ayant pour bornes l'application courante et l'application contenant le secret S, ces applications pouvant être confondues si le secret S existe dans l'application courante.

Pour réaliser l'étape (c), le système de gestion doit effectuer les étapes suivantes consistant à:

(c1) vérifier qu'un registre de sécurité est associé à l'application courante du niveau Ni;

(c2) autoriser la fonction si le registre de sécurité contient le droit requis et terminer la vérification;

5 (c3) rechercher l'existence du secret de référence S au sein de l'application courante de niveau Ni si aucun registre de sécurité n'est associé à l'application courante ou si le registre associé ne contient pas le droit requis;

10 (c4) refuser la fonction et terminer la vérification si le secret existe au sein de l'application courante;

(c5) vérifier qu'un registre de sécurité est associé à l'application parente de niveau $N(i-1)$ de l'application courante si le secret de référence S
15 n'existe pas au sein de l'application courante de niveau Ni;

(c6) autoriser la fonction et terminer la vérification si le registre de sécurité associé à l'application parente contient le droit requis pour
20 utiliser la fonction;

(c7) rechercher l'existence du secret de référence S au sein de l'application parente de niveau $N(i-1)$ de l'application courante si aucun registre de sécurité n'est associé à l'application parente ou si le registre
25 de sécurité associé ne contient pas le droit requis;

(c8) refuser la fonction et terminer la vérification si le secret de référence S existe au sein de l'application parente de niveau $N(i-1)$;

(c9) vérifier qu'un registre de sécurité est
30 associé à l'application grand-parente de niveau $N(i-2)$ de l'application courante suivant l'axe hiérarchique de l'application courante vers l'application racine, si le secret de référence S n'existe pas au sein de l'application parente de niveau $N(i-1)$,

et ainsi de suite tant que l'existence du secret de référence S n'a pas été découvert;

(c10) refuser la fonction et terminer la vérification si le secret n'a pas été découvert.

5 Les figures 4.1 et 4.2 illustrent deux exemples de fonction autorisée tandis que les figures 4.3, 4.4, 4.5 et 4.6 illustrent quatre exemples de fonction refusée.

10 Sur la figure 4.1, la fonction est acceptée car le secret S3 existe en local, et que celui-ci est connu sous le répertoire Rep41.

Sur la figure 4.2, la fonction est acceptée car le secret S1 n'existe pas en local mais que celui-ci est connu sous le répertoire Rep2.

15 Sur la figure 4.3, la fonction est rejetée car le secret S3 existe en local sous le répertoire Rep41 et qu'aucun droit n'a été octroyé sous ce répertoire.

20 Sur la figure 4.4, la fonction est rejetée car le secret S3 existe en local sous le répertoire Rep 41 et que, bien qu'un registre de sécurité R3 soit affecté au répertoire Rep41, la connaissance du secret S3 n'a pas été prouvée.

25 Sur la figure 4.5, la fonction est rejetée car le secret S2, qui n'existe pas en local sous le répertoire Rep41, ni dans le répertoire Rep31, existe sous le répertoire Rep2 et qu'aucun registre de sécurité n'est affecté au répertoire Rep2. Il est à remarquer que la fonction est rejetée bien qu'un secret S2 soit connu sous le répertoire Rep1.

30 Sur la figure 4.6, la fonction est rejetée car le secret S1 n'a pas été trouvé en parcourant l'axe hiérarchique du répertoire Rep41 vers le répertoire Rep1 et ce, bien qu'un secret S1 existe sous les répertoires Rep51 et Rep32.

REVENDEICATIONS

1. Système de gestion de la sécurité d'applications informatiques, caractérisé en ce que:

- les applications informatiques sont enregistrées dans des fichiers de répertoires (Rep1, Rep2, Rep31, Rep32, Rep41, Rep 42, Rep51, Rep52) organisés suivant une arborescence à n niveaux, le répertoire de niveau 1 (Rep1) étant de niveau le plus élevé, et
- un nombre r de registres de sécurité (R) pouvant être affectés chacun à un seul répertoire et chaque registre de sécurité (R) contenant l'ensemble des droits ou secrets S1 à Sp qui ont été octroyés sous un répertoire.

2. Procédé de gestion de la sécurité d'applications informatiques dans un système selon la revendication 1, caractérisé en ce qu'il comprend les étapes suivantes consistant à:

- (a) mémoriser dans des registres de sécurité (R) les droits octroyés (S1 à Sp) sous un répertoire (Rep) selon des règles déterminées (RG1, RG2, RG3),
- (b) rechercher dans l'arborescence le secret présenté, et
- (c) vérifier la connaissance du (ou des) droits au niveau de l'application informatique.

3. Procédé selon la revendication 2, caractérisé en ce que les règles de mémorisation de l'étape (a) sont les suivantes:

- (RG1): attribution d'un registre de sécurité (R) au répertoire courant dès l'octroi d'un

droit sous ce répertoire ou mise à jour
dudit registre de sécurité si un droit a
déjà été octroyé sous ce répertoire,

5 (RG2) perte du lien reliant l'ancien répertoire
courant à son registre de sécurité lors de
la sélection d'un nouveau répertoire sauf
si le répertoire sélectionné est le fils de
l'ancien répertoire courant;

10 (RG3) attribution du registre de sécurité le plus
anciennement attribué au nouveau répertoire
courant si les registres de sécurité sont
tous attribués.

4. Procédé selon la revendication 2 ou 3,
caractérisé en ce que l'étape (b) consiste à appliquer
15 la règle suivante consistant à:

(RG4) vérifier que le secret présenté (S) est
connu dans le répertoire courant (Ni) ou
dans un répertoire de niveau supérieur.

20 5. Procédé selon la revendication 2, 3 ou 4,
caractérisé en ce que l'étape (b) comprend les étapes
intermédiaires suivantes consistant à:

25 (b1) rechercher un secret dans le répertoire
courant de niveau (Ni) et vérifier
l'existence du secret (S) au sein de
l'application,

(b2) si ce secret (S) existe, vérifier que la
présentation du secret est réussie;
= si la présentation est réussie, le droit
associé au secret (S) est octroyé au niveau
30 (Ni) de l'application courante;

- = si la présentation a échoué, le droit associé au secret (S) n'est pas octroyé et la tentative de présentation est terminée;
- (b3) si ce secret (S) n'existe pas dans l'application courante de niveau (Ni), rechercher si ce secret (S) existe au sein de l'application parente de niveau N(i-1).
- (b4) Si ce secret (S) existe dans l'application parente de niveau B(i-1), vérifier que la présentation est réussie:
- = si la présentation est réussie, le droit associé au secret (S) est octroyé dans l'application courante de niveau (Ni),
- = si la présentation a échoué, le droit associé au secret (S) n'est pas octroyé et la tentative de présentation est terminée;
- (b5) si le secret n'existe pas au sein de l'application parente de niveau N(i-1), rechercher l'existence du secret (S) au niveau de l'application de niveau N(i-2) suivant l'axe hiérarchique et vérifier que la présentation est réussie, et ainsi de suite jusqu'au niveau hiérarchique le plus élevé tant que l'existence du secret (S) n'a pas été découvert;
- (b6) Si le secret (S) n'a pas été découvert, la tentative de présentation est terminée.

6. Procédé selon l'une des revendications précédentes 2 à 5, caractérisé en ce que l'étape (c) consiste à appliquer la règle suivante consistant à:

(RG5) Autorisation d'une fonction nécessitant la connaissance d'un secret (S) si et seulement si, en parcourant l'arborescence suivant l'axe hiérarchique de l'application courante vers l'application racine, le premier secret (S) est connu par au moins l'une des applications appartenant à la section arborescente ayant pour bornes l'application courante et l'application contenant le secret (S).

7. Procédé selon l'une des revendications précédentes 1 à 6, caractérisé en ce que l'étape (c) comprend les étapes suivantes consistant à:

- (c1) vérifier qu'un registre de sécurité est associé à l'application courante du niveau N_i ;
- (c2) autoriser la fonction si le registre de sécurité contient le droit requis et terminer la vérification;
- (c3) rechercher l'existence du secret de référence S au sein de l'application courante de niveau N_i si aucun registre de sécurité n'est associé à l'application courante ou si le registre associé ne contient pas le droit requis;
- (c4) refuser la fonction et terminer la vérification si le secret existe au sein de l'application courante;
- (c5) vérifier qu'un registre de sécurité est associé à l'application parente de niveau $N(i-1)$ de l'application courante si le

- secret de référence S n'existe pas au sein de l'application courante de niveau N_i ;
- 5 (c6) autoriser la fonction et terminer la vérification si le registre de sécurité associé à l'application parente contient le droit requis pour utiliser la fonction;
- 10 (c7) rechercher l'existence du secret de référence S au sein de l'application parente de niveau $N(i-1)$ de l'application courante si aucun registre de sécurité n'est associé à l'application parente ou si le registre de sécurité associé ne contient pas le droit requis;
- 15 (c8) refuser la fonction et terminer la vérification si le secret de référence S existe au sein de l'application parente de niveau $N(i-1)$;
- 20 (c9) vérifier qu'un registre de sécurité est associé à l'application grand-parente de niveau $N(i-2)$ de l'application courante suivant l'axe hiérarchique de l'application courante vers l'application racine, si le secret de référence S n'existe pas au sein de l'application parente de niveau $N(i-1)$,
- 25 et ainsi de suite tant que l'existence du secret de référence S n'a pas été découvert;
- 30 (c10) refuser la fonction et terminer la vérification si le secret n'a pas été découvert.

REVENDICATIONS

1. Système de gestion de la sécurité d'applications informatiques, caractérisé en ce que:

- les applications informatiques sont enregistrées dans des fichiers de répertoires (Rep1, Rep2, Rep31, Rep32, Rep41, Rep 42, Rep51, Rep52) organisés suivant une arborescence à n niveaux, le répertoire de niveau 1 (Rep1) étant de niveau le plus élevé, et

- un nombre r de registres de sécurité (R) pouvant être affectés chacun à un seul répertoire et chaque registre de sécurité (R) contenant l'ensemble des droits ou secrets S1 à Sp qui ont été octroyés sous un répertoire.

2. Procédé de gestion de la sécurité d'applications informatiques dans un système selon la revendication 1, caractérisé en ce qu'il comprend les étapes suivantes consistant à:

- (a) mémoriser dans des registres de sécurité (R) les droits octroyés (S1 à Sp) sous un répertoire (Rep) selon des règles déterminées (RG1, RG2, RG3),
- (b) rechercher dans l'arborescence le secret présenté, et
- (c) vérifier la connaissance du (ou des) droits au niveau de l'application informatique.

3. Procédé selon la revendication 2, caractérisé en ce que les règles de mémorisation de l'étape (a) sont les suivantes:

(RG1): attribution d'un registre de sécurité (R) au répertoire courant dès l'octroi d'un

droit sous ce répertoire ou mise à jour
dudit registre de sécurité si un droit a
déjà été octroyé sous ce répertoire,

(RG2) perte du lien reliant l'ancien répertoire
courant à son registre de sécurité lors de
la sélection d'un nouveau répertoire sauf
si le répertoire sélectionné est le fils de
l'ancien répertoire courant;

(RG3) attribution du registre de sécurité le plus
anciennement attribué au nouveau répertoire
courant si les registres de sécurité sont
tous attribués.

4. Procédé selon la revendication 2 ou 3,
caractérisé en ce que l'étape (b) consiste à appliquer
la règle suivante consistant à:

(RG4) vérifier que le secret présenté (S) est
connu dans le répertoire courant (Ni) ou
dans un répertoire de niveau supérieur.

5. Procédé selon la revendication 2, 3 ou 4,
caractérisé en ce que l'étape (b) comprend les étapes
intermédiaires suivantes consistant à:

(b1) rechercher un secret dans le répertoire
courant de niveau (Ni) et vérifier
l'existence du secret (S) au sein de
l'application,

(b2) si ce secret (S) existe, vérifier que la
présentation du secret est réussie;

= si la présentation est réussie, le droit
associé au secret (S) est octroyé au niveau
(Ni) de l'application courante;

= si la présentation a échoué, le droit associé au secret (S) n'est pas octroyé et la tentative de présentation est terminée;

(b3) si ce secret (S) n'existe pas dans l'application courante de niveau (Ni), rechercher si ce secret (S) existe au sein de l'application parente de niveau N(i-1).

(b4) Si ce secret (S) existe dans l'application parente de niveau B(i-1), vérifier que la présentation est réussie:

= si la présentation est réussie, le droit associé au secret (S) est octroyé dans l'application courante de niveau (Ni),

= si la présentation a échoué, le droit associé au secret (S) n'est pas octroyé et la tentative de présentation est terminée;

(b5) si le secret n'existe pas au sein de l'application parente de niveau N(i-1), rechercher l'existence du secret (S) au niveau de l'application de niveau N(i-2) suivant l'axe hiérarchique et vérifier que la présentation est réussie,

et ainsi de suite jusqu'au niveau hiérarchique le plus élevé tant que l'existence du secret (S) n'a pas été découvert;

(b6) Si le secret (S) n'a pas été découvert, la tentative de présentation est terminée.

6. Procédé selon l'une des revendications précédentes 2 à 5, caractérisé en ce que l'étape (c) consiste à appliquer la règle suivante consistant à:

(RG5) Autorisation d'une fonction nécessitant la connaissance d'un secret (S) si et seulement si, en parcourant l'arborescence suivant l'axe hiérarchique de l'application courante vers l'application racine, le premier secret (S) est connu par au moins l'une des applications appartenant à la section arborescente ayant pour bornes l'application courante et l'application contenant le secret (S).

7. Procédé selon l'une des revendications précédentes 2 à 6, caractérisé en ce que l'étape (c) comprend les étapes suivantes consistant à:

- (c1) vérifier qu'un registre de sécurité est associé à l'application courante du niveau N_i ;
- (c2) autoriser la fonction si le registre de sécurité contient le droit requis et terminer la vérification;
- (c3) rechercher l'existence du secret de référence S au sein de l'application courante de niveau N_i si aucun registre de sécurité n'est associé à l'application courante ou si le registre associé ne contient pas le droit requis;
- (c4) refuser la fonction et terminer la vérification si le secret existe au sein de l'application courante;
- (c5) vérifier qu'un registre de sécurité est associé à l'application parente de niveau $N(i-1)$ de l'application courante si le

- secret de référence S n'existe pas au sein de l'application courante de niveau N_i ;
- 5 (c6) autoriser la fonction et terminer la vérification si le registre de sécurité associé à l'application parente contient le droit requis pour utiliser la fonction;
- 10 (c7) rechercher l'existence du secret de référence S au sein de l'application parente de niveau $N(i-1)$ de l'application courante si aucun registre de sécurité n'est associé à l'application parente ou si le registre de sécurité associé ne contient pas le droit requis;
- 15 (c8) refuser la fonction et terminer la vérification si le secret de référence S existe au sein de l'application parente de niveau $N(i-1)$;
- 20 (c9) vérifier qu'un registre de sécurité est associé à l'application grand-parente de niveau $N(i-2)$ de l'application courante suivant l'axe hiérarchique de l'application courante vers l'application racine, si le secret de référence S n'existe pas au sein de l'application parente de niveau $N(i-1)$,
- 25 et ainsi de suite tant que l'existence du secret de référence S n'a pas été découvert;
- 30 (c10) refuser la fonction et terminer la vérification si le secret n'a pas été découvert.

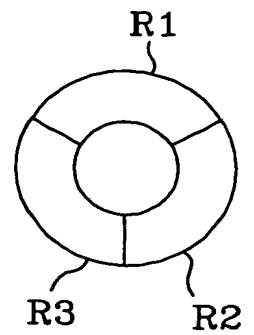
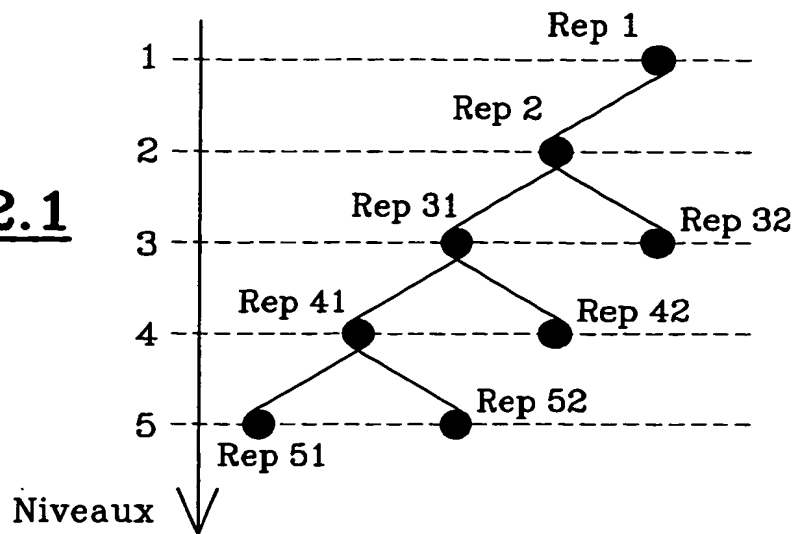
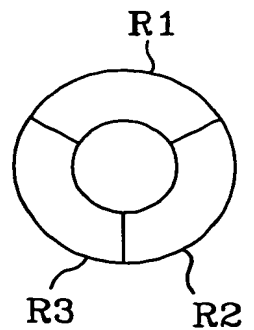
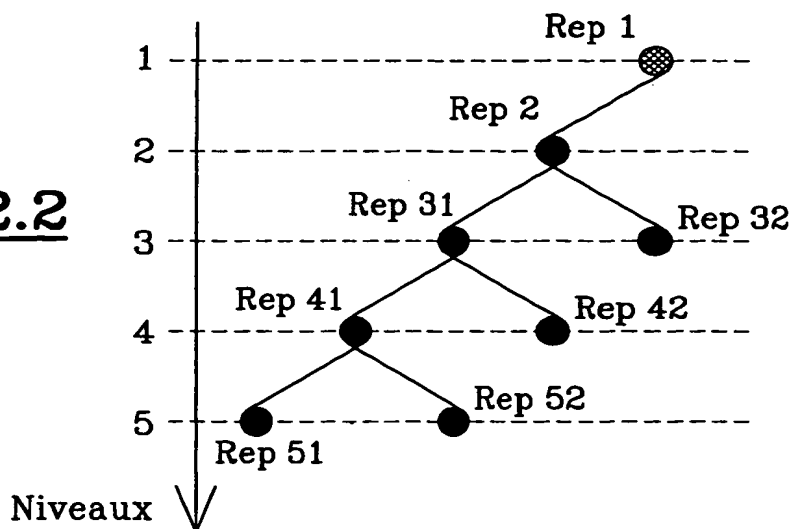
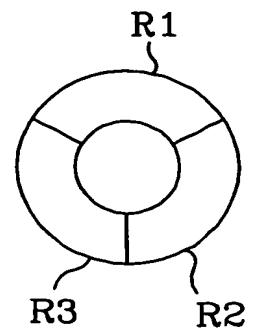
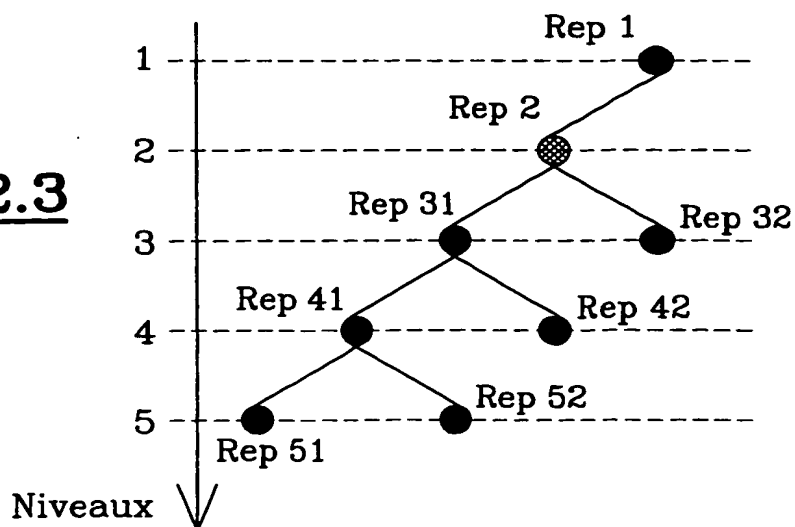
FIG.2.1**FIG.2.2****FIG.2.3**

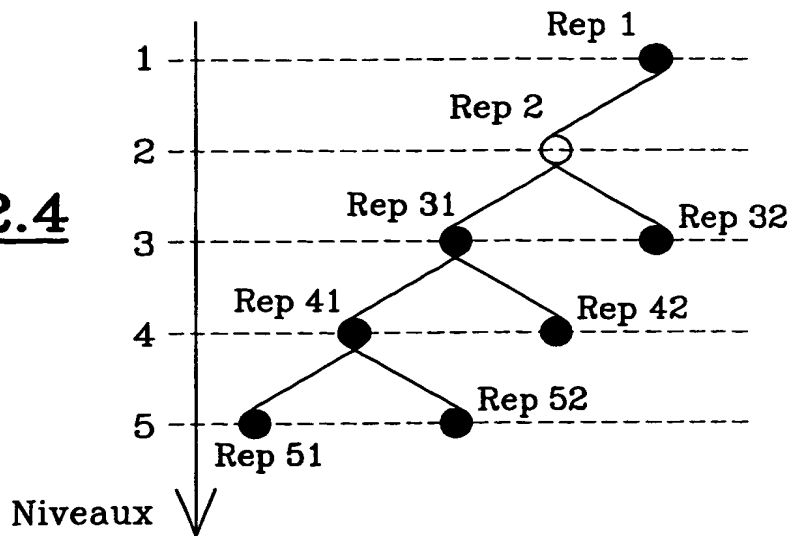
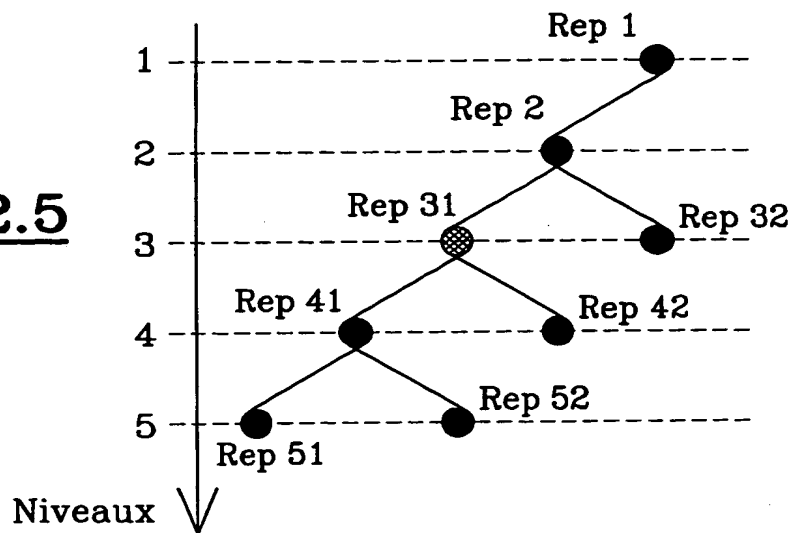
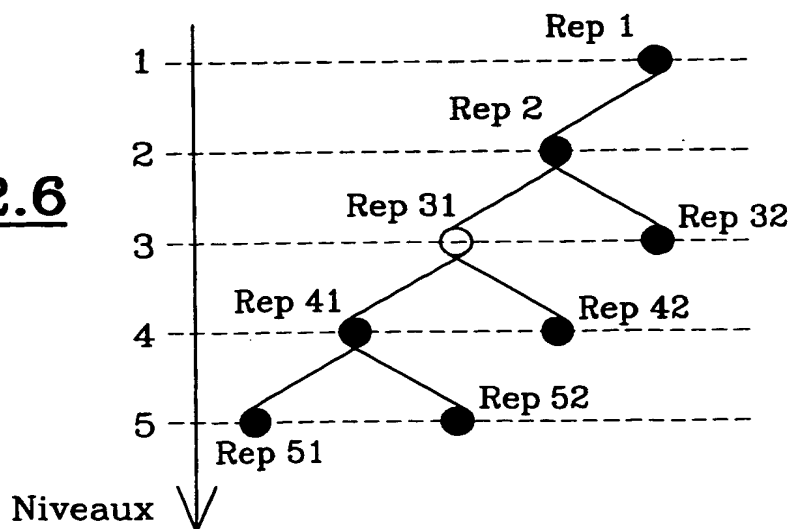
FIG.2.4**FIG.2.5****FIG.2.6**

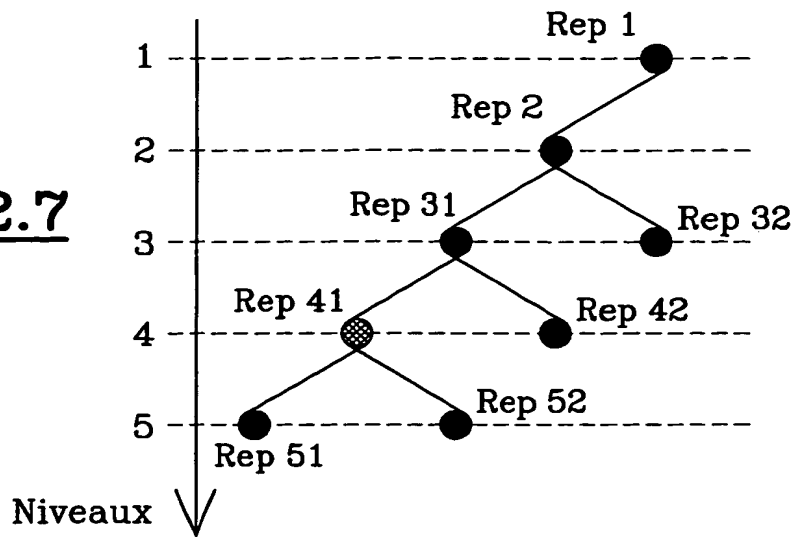
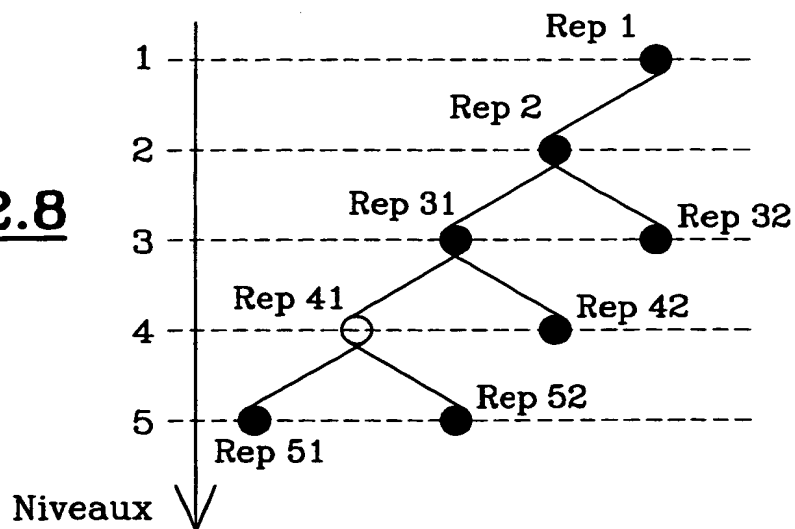
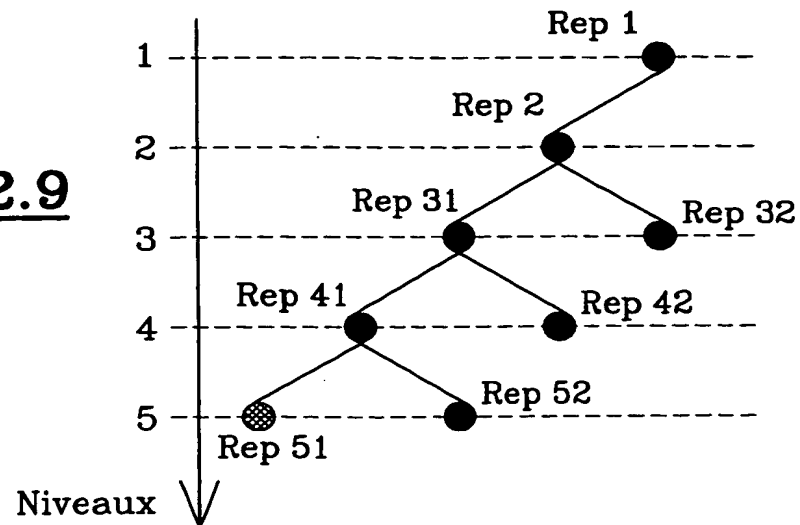
FIG.2.7**FIG.2.8****FIG.2.9**

FIG.2.10

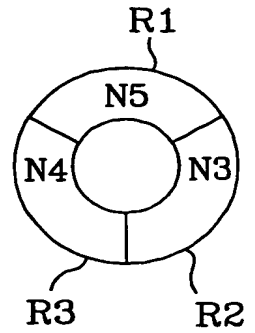
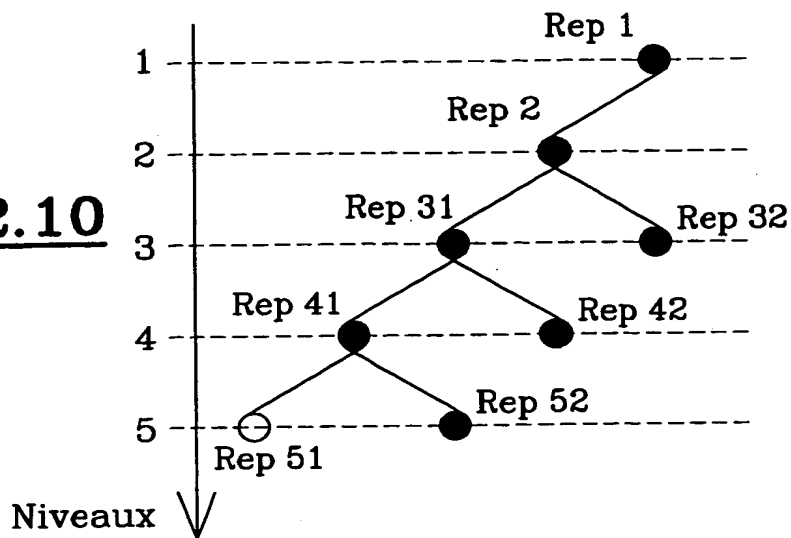


FIG.2.11

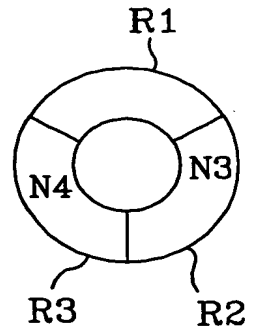
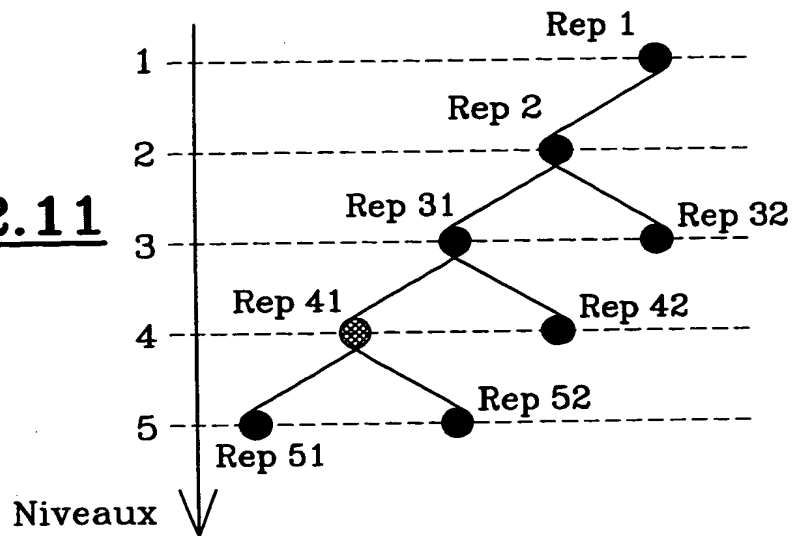


FIG.2.12

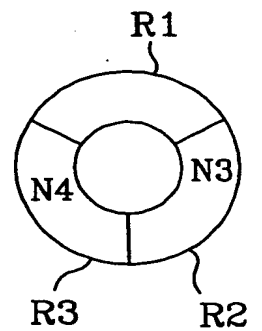
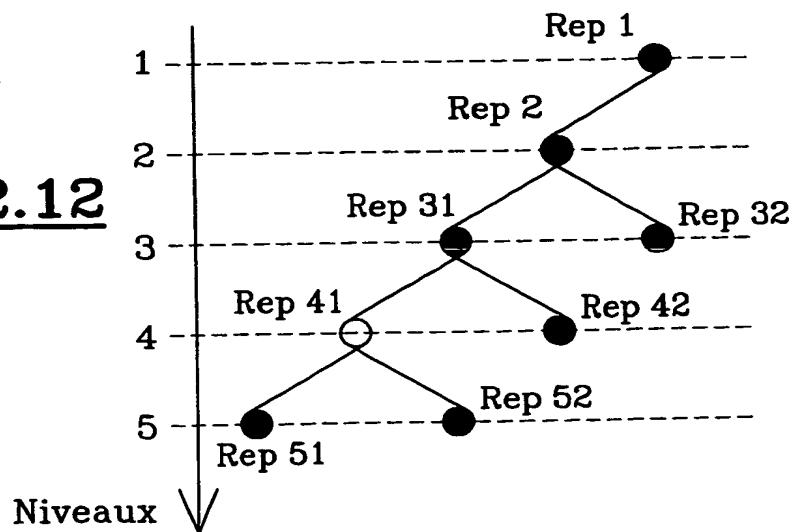


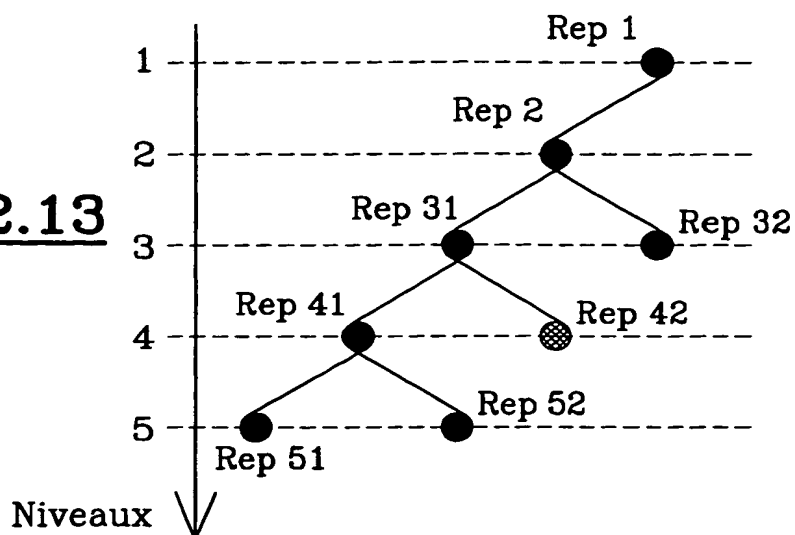
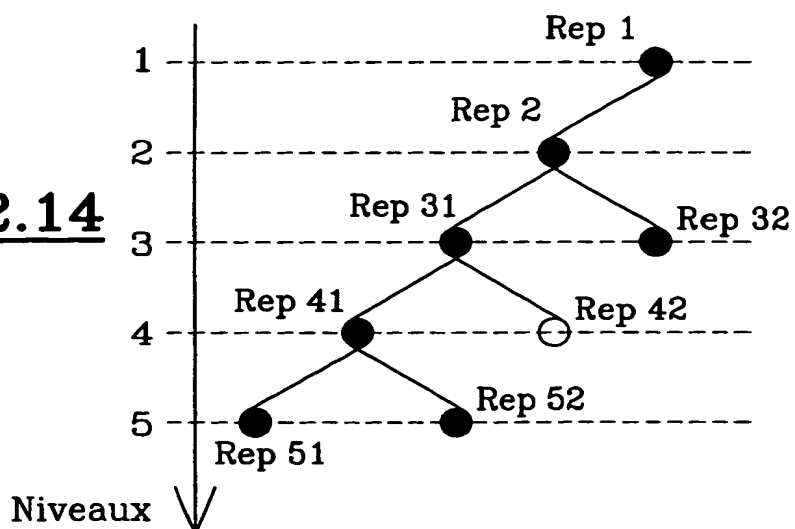
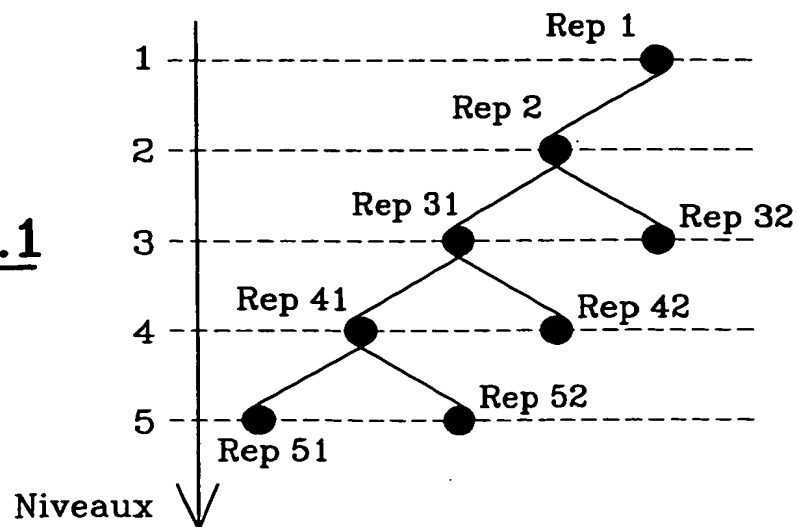
FIG.2.13**FIG.2.14****FIG.1**

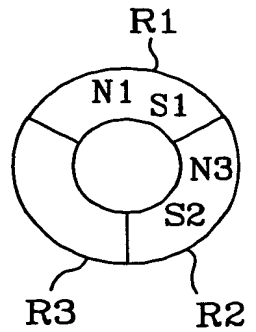
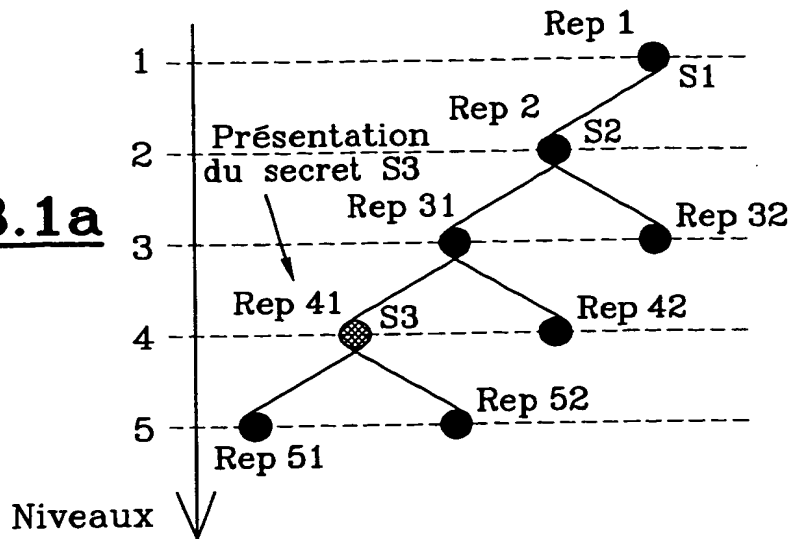
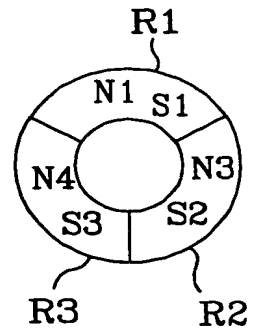
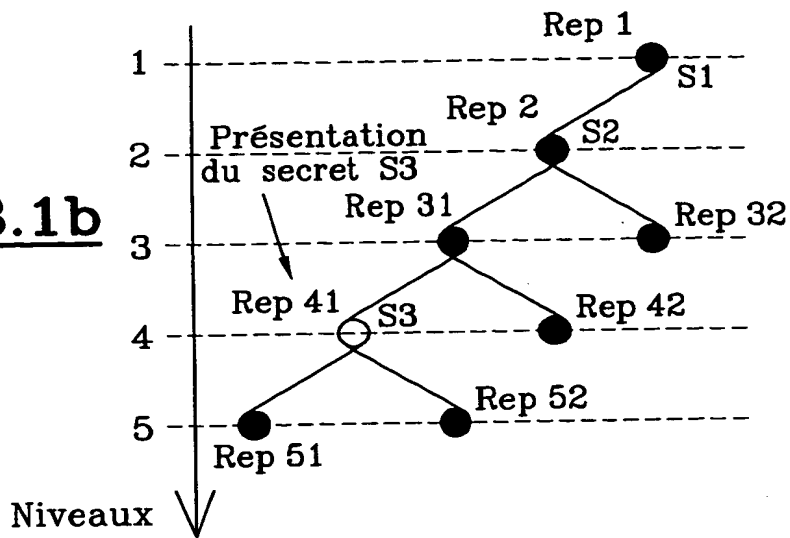
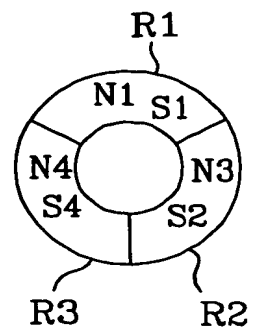
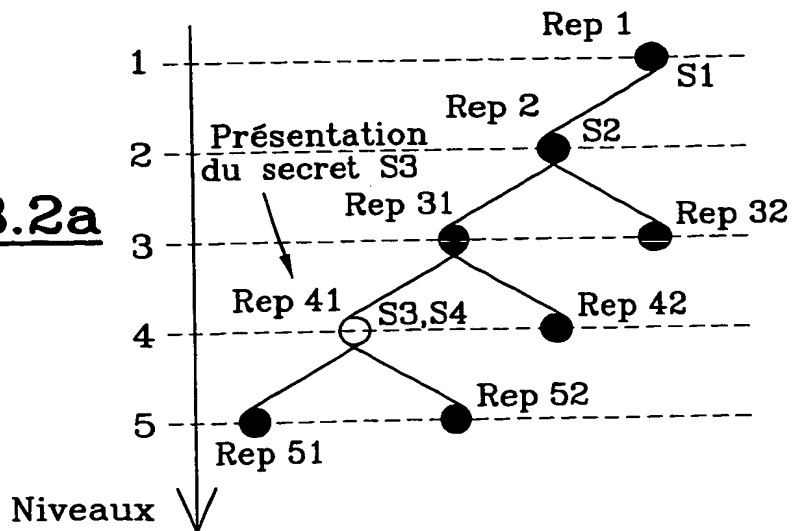
FIG.3.1a**FIG.3.1b****FIG.3.2a**

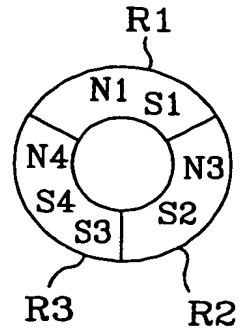
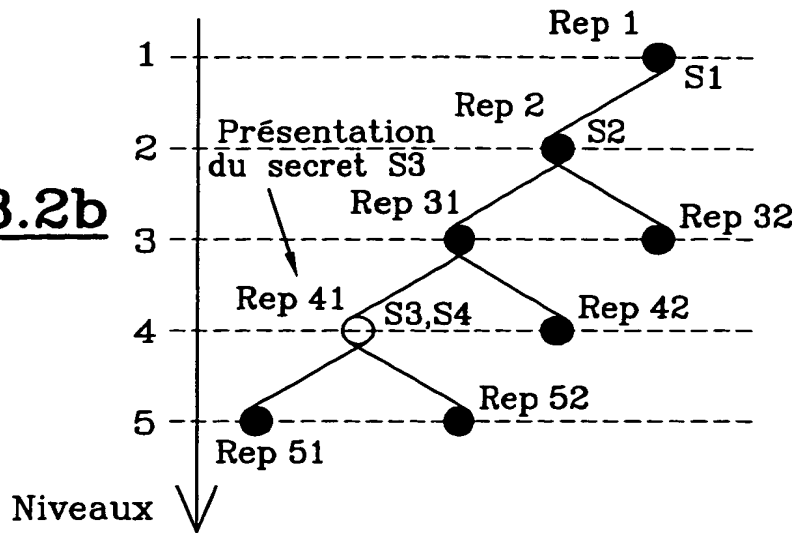
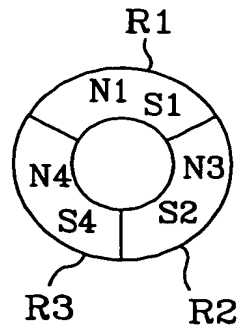
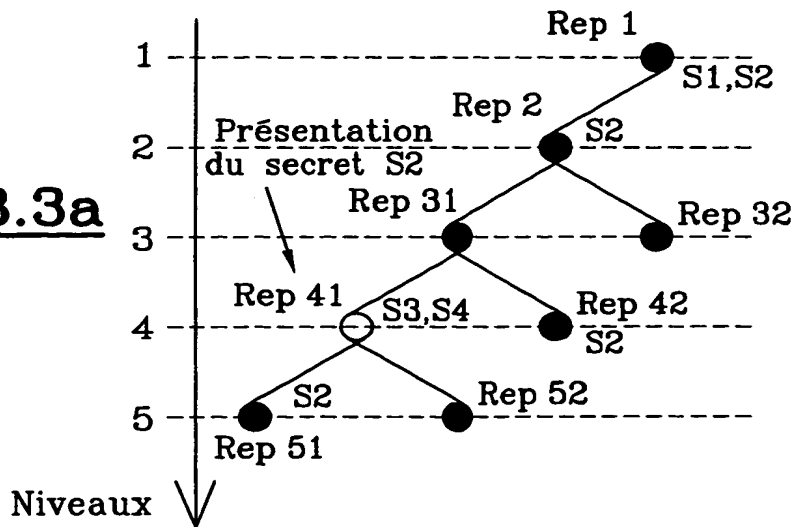
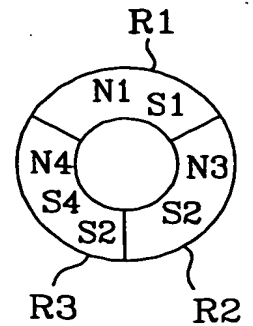
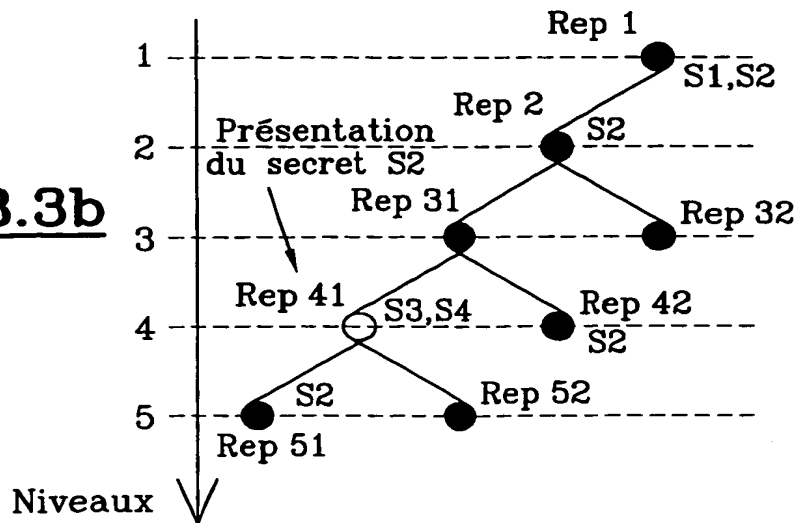
FIG.3.2b**FIG.3.3a****FIG.3.3b**

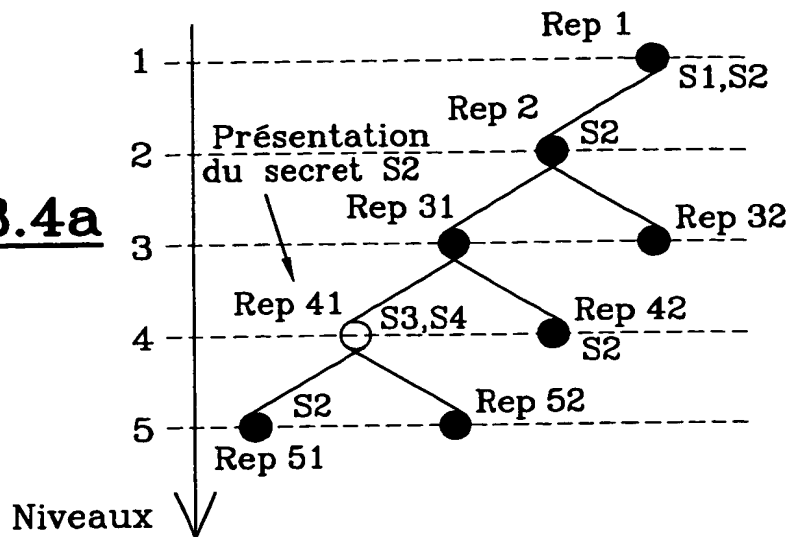
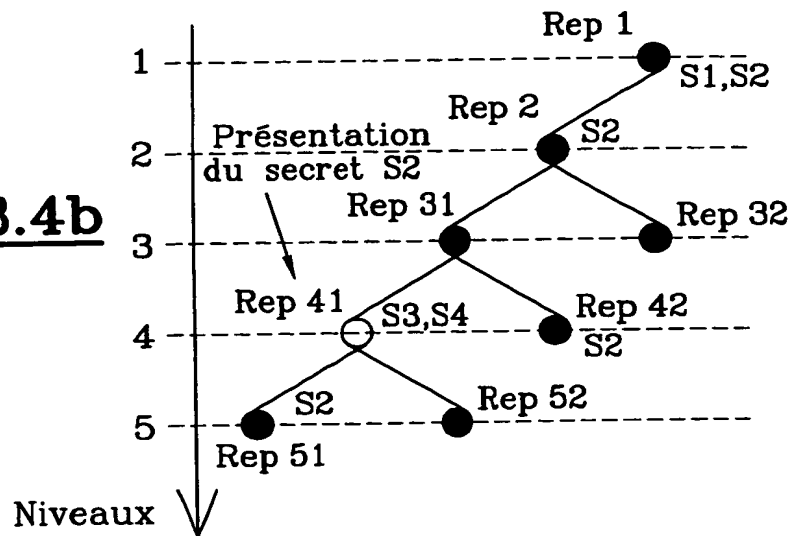
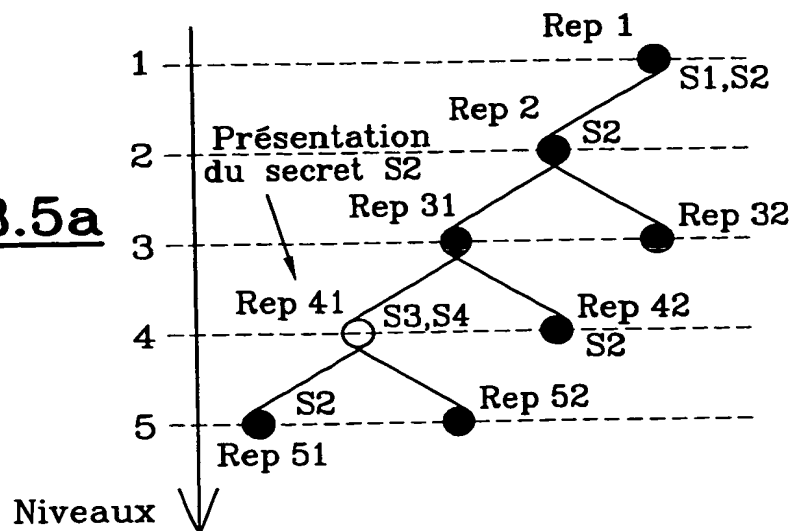
FIG.3.4a**FIG.3.4b****FIG.3.5a**

FIG.3.5b

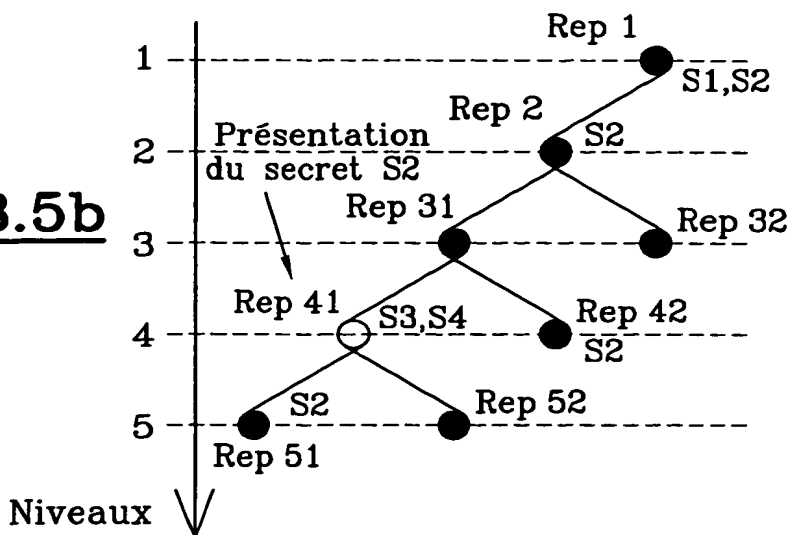


FIG.3.6a

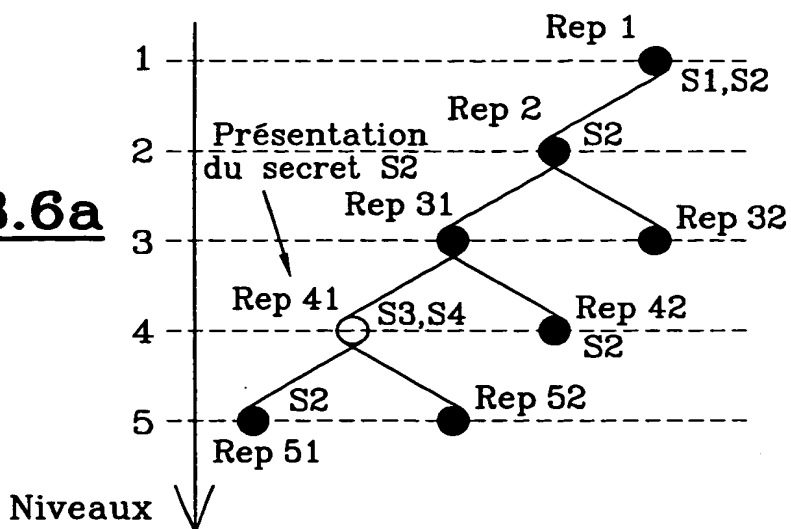


FIG.3.6b

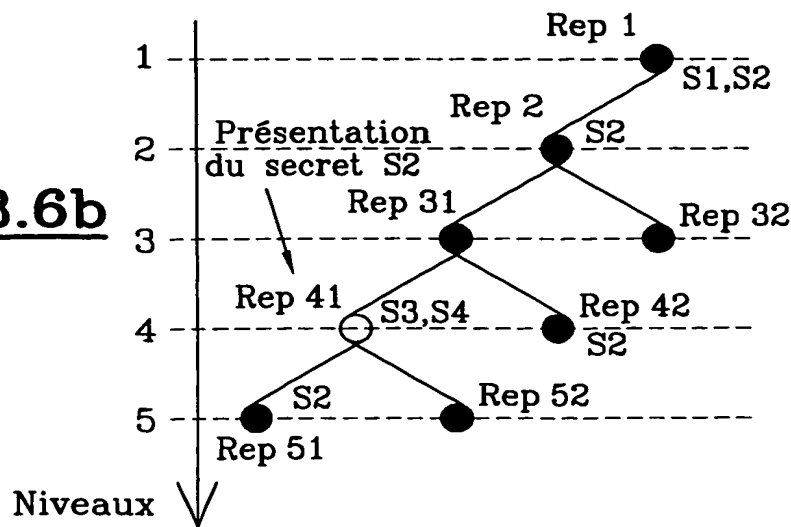


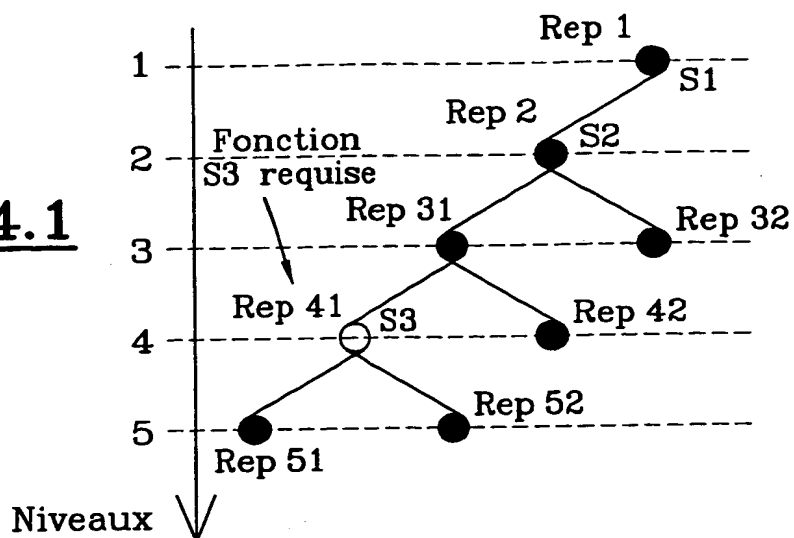
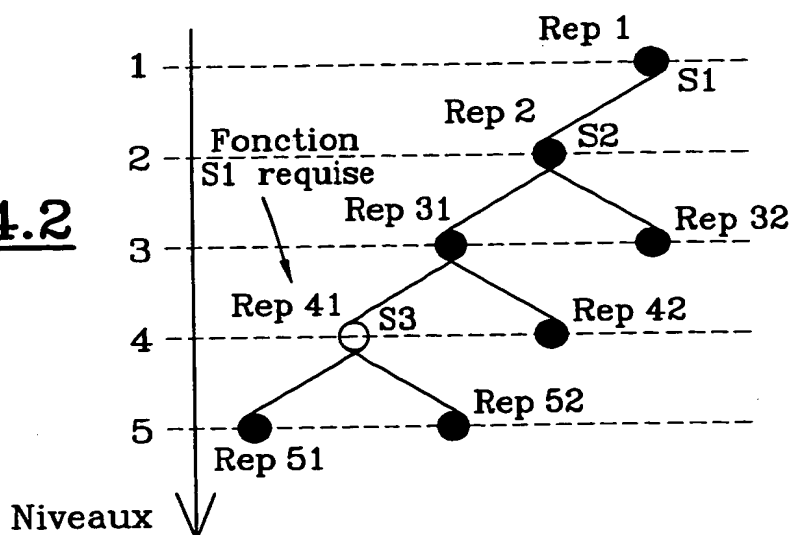
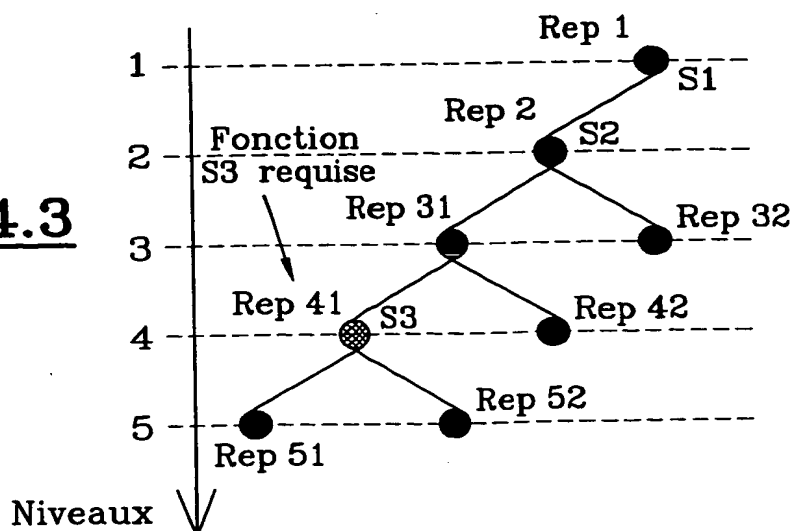
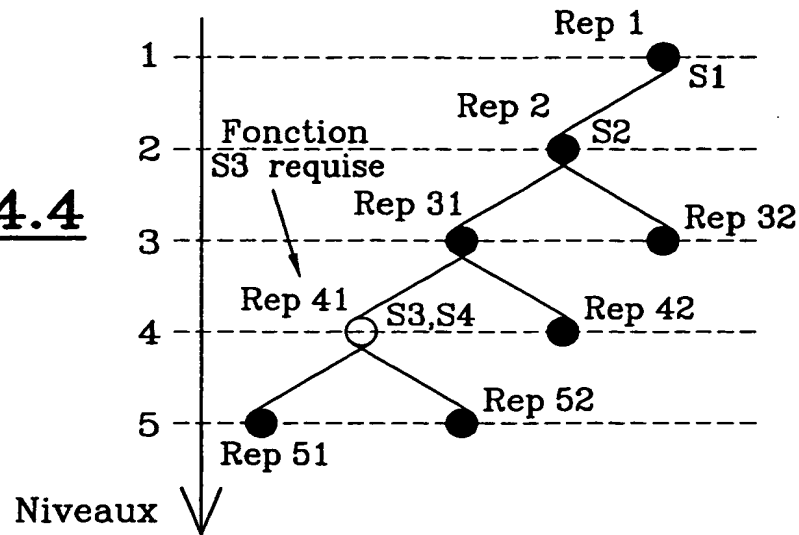
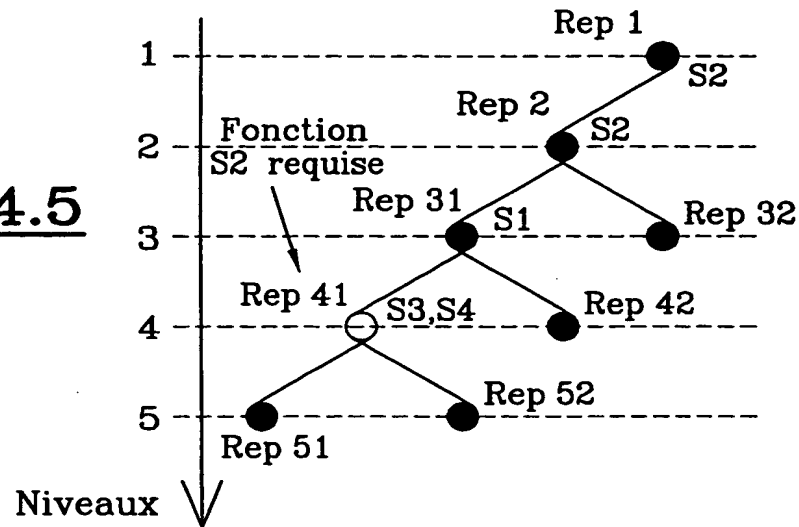
FIG.4.1**FIG.4.2****FIG.4.3**

FIG.4.4**FIG.4.5****FIG.4.6**